

TracNav

- [JPFWiki - Welcome Page](#)
- **[Introduction...](#)**
- **[Installing JPF...](#)**
- **[User Guide](#)**
 - ♦ [Application Types](#)
 - ♦ [JPF Components](#)
 - ♦ [Configuring JPF](#)
 - ♦ [Running JPF](#)
 - ♦ [JPF Output](#)
 - ♦ [The JPF API](#)
- **[Developer Guide...](#)**
- **[Projects...](#)**
- [Summer Projects](#)
- [External Projects](#)
- [Change\(B\)log](#)
- **[About...](#)**
- [Events](#)
- [Presentations](#)
- [Papers](#)
- [FAQ](#)
- [History?](#)
- [Support](#)
- [People?](#)
- [Playground](#)
- [Table of Context](#)

JPF from a User Perspective

This section is where the real fun starts. Here you learn about

- [the basic JPF application types](#)
- [components involved in running JPF](#)
- [how to start JPF](#)
- [how to configure JPF](#)
- [the different types of JPF output](#)
- [how to use JPF from within the test application](#)

All this assumes you are more interested in running JPF than in developing with/for it, so we will leave most of the JPF internal stuff for the developer section? of this wiki.

We do have to bother you with some basic concepts though. Keep in mind that JPF is usually not a black-box tool like - say - a compiler. Most probably you have to configure it according to your needs because

- you have specific verification goals (properties)
- your application has a huge state space that is challenging for a model checker

On the other hand, JPF is also not a "works-or-fails" tool. Depending on how much time you want to invest, you can adapt it to almost all application types and verification goals. And since JPF is open sourced, chances are somebody has already done that